CLOUD DATABASE ROUTE SCHEDULING USING COMBINATION OF PARTICLE SWARM OPTIMIZATION AND GENETIC ALGORITHM

*Shabnam Ghasemi1 and Mohammad Kalantari2
1Department of Computer Engineering, Islamic Azad University, Qazvin Branch, Qazvin, Iran
2Department of Computer Engineering, University, Tehran, Iran
*Author for Correspondence

ABSTRACT
In cloud computing environment, accessibility of user to the cloud database is a challenging process that needs a route scheduling scheme to manage the accessibility of users. For this purpose, combination of Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) is proposed for route scheduling. Results show that the proposed hybrid PSO-GA is superior to the other similar approaches for cloud database route scheduling.

Keywords: Cloud Computing, Cloud Database, Genetic Algorithm, Particle Swarm Optimization Algorithm

INTRODUCTION
One of the services provided in the Internet environment is sharing the computing resources in order to handle applications. The word “computing resources” refers to servers, storages, data, application and etc. which are shared and accessible through the Internet (Paquette, 2010). To share these resources, some central remote servers are needed to storage the data and applications. This type of computing can be renamed to cloud computing where the consumers can store and access to data, application and program over the Internet rather than the local computer’s hard drive (Zhang, 2010). For instance, Yahoo Email, Gmail, Google Drive and etc. are the simple examples of cloud computing that the servers and user interface applications are all on the cloud (Internet). Cloud computing is divided into three different parts: application, connection and storage. Between these three parts, storage is one of the important issues that is addressed in this paper. Resources of the cloud computing are stored and retrieved to/from cloud databases. Cloud databases should be big enough to store resources which contain probably high volume of data or applications. It is important that the data volume accessed by the users on the web is growing and so, the volume of databases should be increased. The main key issue concern to cloud databases is simultaneously accessibility of users to the cloud databases. In this regard, accessibility of users to the cloud databases should be managed using route scheduling in order to speed up the accessibility of consumers to the cloud databases (Divyakant, 2010). In this paper, a new method of cloud database route scheduling is proposed which is based on heuristic approaches. In this regard, combination of Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) will be presented in this paper.

This paper is organized as the follows; the cloud database is described in Section 2. Section 3 focuses on the problem definition. The proposed method will be presented in Section 4. Analyzing of the proposed method is performed in Section 5 and the paper is concluded in Section 6.

Cloud Database
In a cloud database system, there are some nodes which are connected though the network connection. Each node has its own processor, database and database management system. In the other word, a cloud database system consists of several centralized database units which show union set of physically distributed cloud databases (YanHua, 2011). Nowadays, sharing and requesting information, data and application on the web are interested by the users. It should be considered that the volume of the shared data on the web is growing and so, two main challenges are faced: storage capacity and accessibility management. Storage capacity of cloud databases should be big enough to save the data which is shared by the consumers on the web. Accessibility
management is used when the requests of several users are transmitted through the same route (Daniel, 2009). Simultaneously transmission of the requests on the same route to the same cloud database increases the probability of missing the data or obtaining the data with high delay and so, the quality of services to is decreased. To solve the problem, accessibility of the users to the cloud databases should be managed using the route scheduling scheme. Route scheduling is the plan of finding the optimal route to the best cloud database in order to service the users with the reduced delay. In cloud database route scheduling, some factors should be regarded: congestion control, network flow distribution, routing list and priority management (Mateljan, 2010).

Characteristic of cloud databases is dynamically distribution and updating of databases. Thus tree search algorithm and static search algorithm are inappropriate methods for route scheduling. In contrast, meta-heuristic algorithms, such as Particle Swarm Optimization (PSO)(Kennedy, 1995) Genetic Algorithm (GA)(Holland, 1992), Ant Colony Optimization (ACO)(Shi, 2010) and etc. are the best choice for the route scheduling because of their capability to deal with large scale problems and finding optimum solution in the reduced runtime. In (Shi, 2010), cloud database route scheduling is performed using ACO approach and so, smart routing, speedy access to the cloud databases and global optimization of distributed computing are achieved. This research work suffers from the problems of insufficient amount of pheromones produced by the artificial ants at the initial steps. So, following of initial produced pheromones by the ants takes long time. In (Yanhua, 2011) the problem of previous work is solved using the combination of ACO and GA (ACO-GA) where the GA function is employed to generate extra pheromone.

In (Guo, 2012), the problem of distribution of cloud databases is analyzed and its reasons comeback to high volume of data and low bandwidth. To achieve this problem, the PSO-based approach is proposed.

**Problem Definition**

Since the proposed method uses the classes of meta-heuristic algorithms, several algorithms which are included in this class are described as follows.

In swarm intelligence algorithms, a group of particles is defined in order to find the global best point of problem. To find the best point, each particle explores the search space and informs the other particles from its own search results. The characteristics of swarm intelligence algorithms make them suitable for optimization problems, like scalability, fault tolerance, consistency, speedy search, flexibility and parallelism. Some of the well-known methods of swarm intelligence are Particle Swarm Optimization (PSO)(Kennedy, 1995) Genetic Algorithm (Holland, 1995) and Ant Colony Optimization (ACO) (Dorigo, 2006).

In this paper, two meta-heuristic-based methods are improved for cloud databases route scheduling: Particle Swarm Optimization (PSO) and Genetic Algorithm (GA).

3.1 Particle Swarm Optimization (PSO)

PSO (Kennedy, 1995) is based on the animals’ social interactions such as bird or fish swarms. In this method, a swarm of particles is defined that each of particles searches for the final best solution based on the best personal experience and also the best swarm experience.

To model the PSO, a population of N particles in D-dimension space is defined. For each particle, two vectors of position \( x_i = (x_{i1}, x_{i2}, \ldots, x_{iD}) \) and velocity \( v_i = (v_{i1}, v_{i2}, \ldots, v_{iD}) \) are defined. The two vectors are updated according to (1) and (2), respectively:

\[
\begin{align*}
    v_{i,j}(t + 1) &= w v_{i,j}(t) + c_1 R_{i,j}^1 \left( P_{best,i,j}(t) + x_{i,j}(t) \right) + c_2 R_{i,j}^2 \left( G_{best,i,j}(t) + x_{i,j}(t) \right) \\
    x_{i,j}(t + 1) &= x_{i,j}(t) + v_{i,j}(t + 1)
\end{align*}
\]

where, \( x_{i,j} \) is the component \( j \) of particle \( i \), \( c_1 \) and \( c_2 \) are acceleration coefficients and \( w \) is a constant value for inertia weight and \( R \) is a random number with uniform distribution in \([0, 1]\). \( P_{best}(t) \) refers to the best
Research Article

position of \(i^{th}\) particle until time \(t\) (the best individual experience of particle \(i\)) and \(Gbest\) \((t)\) is the best position found by the whole particles (the best swarm experience). At each iteration, the \(Pbest_i\) and \(Pbest\), are updated by (3) and (4), respectively:

\[
Pbest_i(t + 1) = \begin{cases} 
Pbest(t), & \text{if } f(x_i(t + 1)) \geq f(Pbestx_i(t)) \\
x_i(t), & \text{if } f(x_i(t + 1)) < f(Pbestx_i(t))
\end{cases} \tag{3}
\]

\[
Gbest(t + 1) = \underset{1 \leq i \leq N}{\operatorname{argmin}} f(Pbest_i(t + 1)) 
\tag{4}
\]

where \(f(x)\) is the fitness value of vector \(x\). Note that the parameter \(\operatorname{argmin}_{p_i}\) can be replaced by \(\operatorname{argmax}_{p_i}\) according to the optimization problem.

3.2 Genetic Algorithm (GA)

GA (Holland, 1995) belongs to a set of evolutionary algorithms and based on the process of natural selection. In the other word, GA heuristically searches to find the best solutions for optimization and search problems (Holland, 1995).

GA proposes solutions in the form of chromosomes. In evolutionary process of GA, there are four main stages: parent selection, crossover, and mutation and survival selection. In the first step of GA cycle, the two best chromosomes of \(k\) randomly selected chromosomes are picked out using tournament selection of size \(k\) (Holland, 1995). After that, the crossover operator with type of “Uniform” and probability of \(Pc\) is applied to the chromosomes to generate offspring. Next, mutation operator with type of “Bitwise bit-flipping” and probability of \(Pm\) is utilized to apply small changes on the chromosomes with the probability of \(Pm\) which controls the exploitation rate. Note that increasing of exploration rate decreases the similarity between the offspring and their parent while low exploration rate in initial optimization step may lead the population to the local optima.

Proposed Method

In this Section, the details of the proposed method will be described. The innovation of this study is combination of PSO and GA with interest to improving of the PSO. For this purpose, two different models of this combination are proposed: PSO with mutation operator and PSO with crossover operator.

PSO with Mutation Operator

The main problem of PSO is falling into local optima. To solve this problem and escaping from the local optima, it may be effective to employ the mutation operator into optimization process of PSO. For this purpose, a probability value is determined for each variable of a particle.

If a variable of particle is probably selected to mutate, the value of the variable is changed according to the type of mutation which are described as follows. If the fitness value of mutated particle is better than the main particle’s fitness value, the mutated particle is replaced with its main particle (parent) otherwise the main particle is remained.

To perform the mutation operator, there are three ways:

- If representation of particles is binary, simple mutation is employed whicha variable of particle with the value of 0 changes to 1 and vice versa.
- If representation of particles is a set of real numbers, the uniform mutation is applied where a random number from a given interval is generated in order to replace with the value of the selected variable.
- The last operator of mutation is dynamic mutation that is formulated as:

\[
x_k' = x_k + \Delta(t, x_k^U - x_k^L) 
\]

where \(x_k^U\) and \(x_k^L\) are the lower and upper bounds of \(x\).
PSO with Crossover Operator

The second model for the combination of PSO and GA is development of crossover into PSO. A simple crossover operator with breeding approach is set into the PSO procedure. In breeding scheme, following steps should be passed:

- Computation of the position and velocity for each particle.
- Assigning a breeding probability value, \( P_b \), to each particle.
- Choosing two chromosomes \( x_a \) and \( x_b \) as the parent according to their probabilities rather than their fitness values (characteristic of breeding scheme) in order to generate two offspring. The positions of offspring, \( x'_a(t) \) and \( x'_b(t) \), are computed as the following equation

\[
\begin{align*}
x'_a(t) &= r x_a(t) + (1 - r) x_b(t) \\
x'_b(t) &= r x_b(t) + (1 - r) x_a(t)
\end{align*}
\]

where \( r \) is a random value inside the interval \([0,1]\) and \( t \) refers to the current generation. The velocities of offspring are formulated as:

\[
\begin{align*}
v'_a(t) &= \frac{v_a(t) + v_b(t)}{\|v_a(t) + v_b(t)\|} \times \|v_a(t)\| \quad (6) \\
v'_b(t) &= \frac{v_a(t) + v_b(t)}{\|v_a(t) + v_b(t)\|} \times \|v_b(t)\| \quad (7)
\end{align*}
\]

where \( v_a(t) \) and \( v_b(t) \) are the velocity of parent and \( \| . \| \) means norm distance.

- Replacing of offspring to their parent if the fitness value of offspring is better than their parent.

At follows, several strategies are proposed for applying the simple crossover on to particles:

First strategy: after the computation of positions and velocities, particles are ranked according to their fitness (PSO function). At the next step, several strong and weak particles are selected for the crossover step and offspring generation (GA function with just crossover operator). After that, the velocity and position of the generated offspring are calculated using (6)-(9).

Second strategy: in this idea, 50% of whole particles are chosen for the crossover step (GA function with just crossover operator) and the other particles are processed using the PSO function. Note that the position and velocity of the generated offspring are calculated in GA function using (6)-(9).

Third strategy: in this strategy, first the PSO function is applied on the whole population. After that the updated population is passed into GA function (with just crossover operator). It is important that the position and velocity of each generated offspring inside the GA function is calculated using (6)-(9).

Last strategy: this strategy is focused on the length of particles that 50% of variables of whole population are optimized using the PSO and the other variables of all particles are passed into GA function to generate the offspring. It is important that the position and velocity of the offspring is computed inside the GA function using (6)-(9).

The characteristics of the proposed PSO-GA are listed as follows:

High: Convergence Rate: this feature refers to the advantage of PSO where the updating formula of particle’s velocity is an effective factor to control and speeding of the convergence rate.

Full Search Capability: this property is inherited from the GA where the proposed method is able to perform full search on the problem space to find the global optimum.

Stability: the feature of meta-heuristic algorithms is their randomness behavior. A meta-heuristic algorithm is stable if it can almost always find out the global optimum using its randomness behavior. Experimental results prove the stability of the proposed method.

RESULTS AND DISCUSSION

In this section, the proposed hybrid PSO-GA is analyzed on the cloud database routing. In this step, Cloud Sim (CLOUDS) framework is employed to simulate the cloud databases. In the Cloud Sim,
different configurations of testing system are classed to the different groups which are mentioned in Table 1. Note that the testing systems are cloud databases which are distributed randomly.

To model the testing environment, a connected graph of \( n \) nodes is considered and formulated as \( G = (V, E) \) that \( V \) is the set of nodes and \( E \) shows the connections between the nodes. Problem of cloud database route scheduling is equivalent to finding the shortest path between the user and cloud database and is formulated as (Hua, 2011):

PSO (Guo, 2012), GA (Medina, 2001), ACO (Ant Colony Optimization) (Shi, 2010) and ACO-GA (Combination of Ant Colony Optimization and Genetic Algorithm) (Hua, 2011). Results of the proposed method which are compared with the similar meta-heuristic algorithms are presented in Figure 2. According to this figure, the results are demonstrated for the groups of A, B and C with the goal of finding the shortest path.

Note that in this figure, for each group, the average values of all hybrid strategies (reported in Figure 1) are considered for the proposed method. Figure 2 proves the superiority of the proposed hybrid PSO-GA to the other methods because of the stability feature of the proposed method. According this figure, the second best method is PSO because of its high convergence rate and stability. The results of ACO-GA are better than the simple GA because of the reason mentioned in Section 3. The worst results are obtained using the GA approach because of its low convergence rate. In Fig. 3, the proposed method and the other meta-heuristic algorithms are analyzed on the user access time.

**Conclusion**

In this paper, the problem of cloud database route scheduling is analyzed. To solve the problem, a new combination of PSO and GA with different strategies is proposed in order to find the shortest path between the user and cloud database.
The result of finding the shortest path is reducing of user accessibility time to the databases. The proposed hybrid PSO-GA is analyzed on finding the shortest path and accessibility time and the results of the comparison shows the priority of the proposed method to similar meta-heuristics algorithms.

Table 1: Comparison of hybrid strategies on user’s access time (in second unit)

<table>
<thead>
<tr>
<th>Group</th>
<th>First strategy</th>
<th>Second strategy</th>
<th>Third strategy</th>
<th>Last strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>11</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>B</td>
<td>18</td>
<td>21</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>29</td>
<td>22</td>
<td>45</td>
</tr>
<tr>
<td>D</td>
<td>56</td>
<td>61</td>
<td>49</td>
<td>85</td>
</tr>
<tr>
<td>E</td>
<td>105</td>
<td>120</td>
<td>89</td>
<td>195</td>
</tr>
<tr>
<td>F</td>
<td>215</td>
<td>235</td>
<td>150</td>
<td>540</td>
</tr>
</tbody>
</table>

Figure 2: Comparison of the proposed hybrid PSO-GA with the similar meta-heuristic algorithms on finding the shortest path to cloud database

Figure 3: Comparison of the proposed hybrid PSO-GA with the similar meta-heuristic algorithms on user access time to cloud database
REFERENCES


